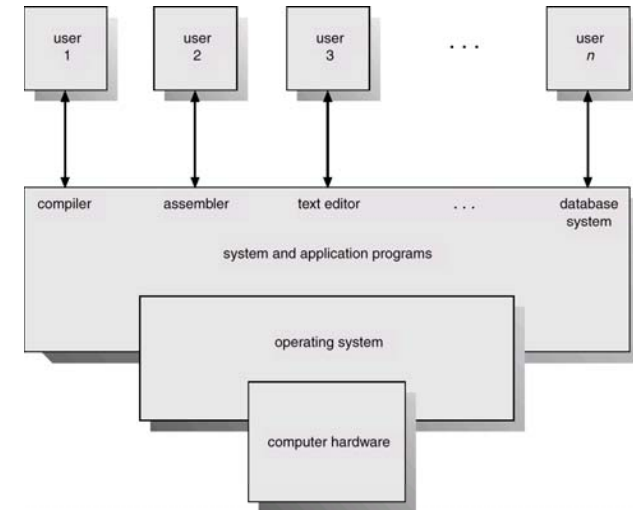


컴퓨터 시스템의 기본 구조



컴퓨터시스템

컴퓨터시스템 개요

컴퓨터 시스템의 구성 요소

1. 하드웨어

- 물리적인 컴퓨팅 자원
- (ex) CPU, 메모리, 입출력장치

2. 운영체제

- 응용프로그램들의 하드웨어 사용을 제어하고 조정하고, 사용자들에게 컴퓨터 사용을 위한 인터페이스를 제공함
- (ex) UNIX, Linux, Windows XP, Mac OS

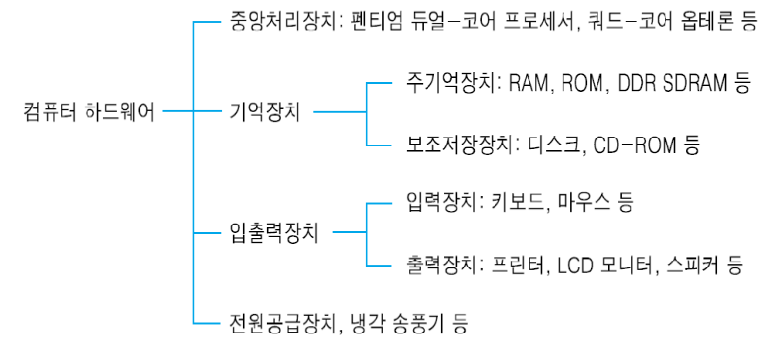
3. 응용 프로그램

- 사용자들의 문제를 해결하기 위한 프로그램(소프트웨어)
- (ex) 워드프로세서, 컴파일러, 웹브라우저, 데이터베이스시스템, 게임, 스프레드시트(엑셀)

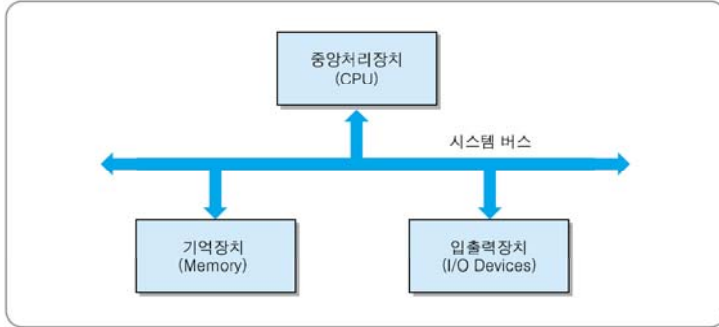
4. 사용자

- 사람, 기계, 다른 컴퓨터

컴퓨터 하드웨어의 주요 구성요소



■ 기본 구조



- 기본 기능: 프로그램 코드를 정해진 순서대로 실행하는 것
 - 필요한 데이터를 읽어서(read), 처리(processing)하고, 저장(store)한다.

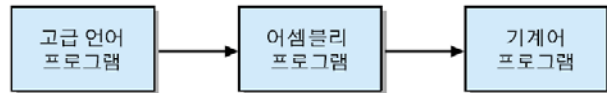
- 중앙처리장치(Central Processing Unit: CPU)
 - 프로세서(processor) 라고도 함
 - '프로그램 실행'과 '데이터 처리'라는 중추적 기능의 수행을 담당
- 기억장치(memory)
 - (1) 주기억장치(main memory)
 - CPU 가까이 위치하며, 반도체 기억장치 칩들로 구성
 - 고속 액세스
 - 가격이 높고 면적을 많이 차지 → 저장 용량의 한계
 - 영구 저장 능력이 없기 때문에, 일시적 저장장치로만 사용 (휘발성 메모리)

- (2) 보조저장장치(auxiliary storage device)
 - 2차 기억장치(secondary memory)
 - 기계적인 장치가 포함되기 때문에 저속 액세스
 - 저장 밀도가 높고, 비트 당 비용이 저가
 - 영구 저장 능력을 가진 저장장치 : 디스크, 자기 테이프 등
- 입출력장치(I/O device)
 - 입력 장치(input device), 출력 장치(output device)
 - 각 장치와 CPU간의 데이터 이동을 위한 별도의 **제어기**가 있음
 - 사용자와 컴퓨터간의 대화를 위한 도구

- 컴퓨터 정보:
 - 2진수 비트(1과 0)들로 표현된 프로그램 코드와 데이터
- 프로그램 코드
 - 기계어(machine language)
 - 컴퓨터 하드웨어 부품들(주로 CPU)이 이해할 수 있는 언어로서, 2진 비트들로 구성
 - 어셈블리 언어(assembly language)
 - 고급 언어와 기계어 사이의 중간 언어,
 - 저급 언어(low-level language), 기계어와 1:1 대응
 - 기계어 코드를 사람이 이해할 수 있는 심볼로 나타내어 사용
 - 고급 언어(high-level language)
 - 영문자와 숫자로 구성되는 사람이 이해하기 쉬운 언어
 - C, C++, Java, PASCAL, FORTRAN, COBOL 등

프로그램 언어의 변환 과정

- [예] $Z = X + Y$
 - LOAD A, X : 기억장치 X번지의 내용을 읽어 레지스터 A에 적재 (load) ($A \leftarrow M[X]$)
 - ADD A, Y : 기억장치 Y번지 내용을 읽어 레지스터 A에 적재된 값과 더하고, 결과를 다시 A에 적재 ($A \leftarrow A + M[Y]$)
 - STOR Z, A : 그 값을 기억장치 Z 번지에 저장(store) ($M[Z] \leftarrow A$)



[예] $Z = X + Y$

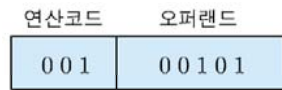
LOAD A, X	00100101
ADD A, Y	10000110
STOR Z, A	01000111

프로그램 언어 번역 소프트웨어

- 컴파일러(compiler)
 - 고급언어 프로그램을 기계어 프로그램으로 번역하는 소프트웨어
- 어셈블러(assembly)
 - 어셈블리 프로그램을 기계어 프로그램으로 번역하는 소프트웨어
 - 니모닉스(mnemonics)
 - 어셈블리 명령어가 지정하는 동작을 개략적으로 짐작할 수 있도록 하기 위하여 사용된 기호
 - (ex) 'LOAD', 'ADD', 'STOR' 등

기계 명령어의 형식

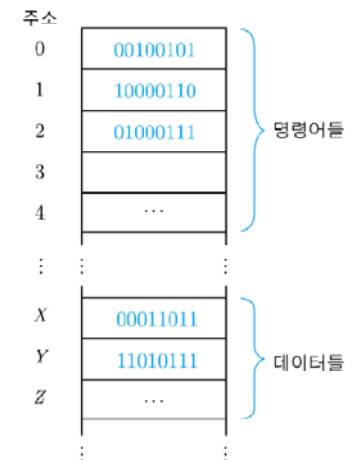
- 기계 명령어(machine instruction) 형식



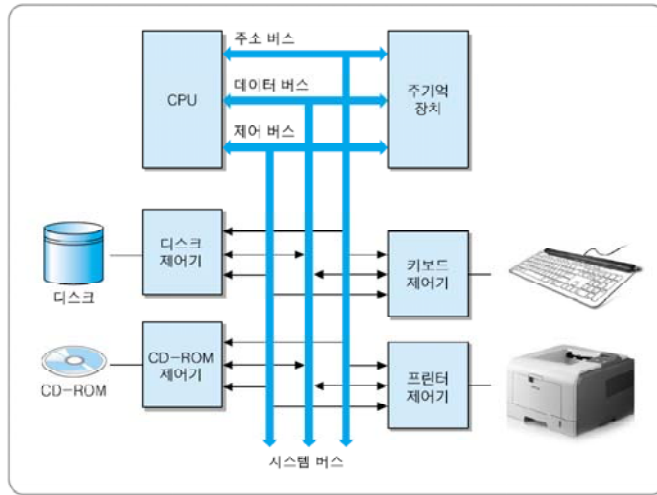
- 연산 코드(op code)
 - CPU가 수행할 연산을 지정해 주는 비트들
 - 비트 수 = 3이면, 지정할 수 있는 연산의 최대 수는 $2^3 = 8$
- 오퍼랜드(operand)
 - 적재/저장되거나 연산에 사용될 데이터가 저장된 기억장치 주소
 - 비트 수 = 5 라면, 주소지정(addressing)할 수 있는 기억 장소의 최대 수는 $2^5 = 32$ 개

프로그램 코드와 데이터의 기억장치 저장

- 워드(word)
 - 각 기억 장소에 저장되는 데이터의 기본 단위로써, CPU에 의해 한 번에 처리될 수 있는 비트들의 그룹
 - [예] 8-비트, 16-비트, 32-비트, 혹은 64-비트 워드
- 바이트(byte)
 - 8-비트

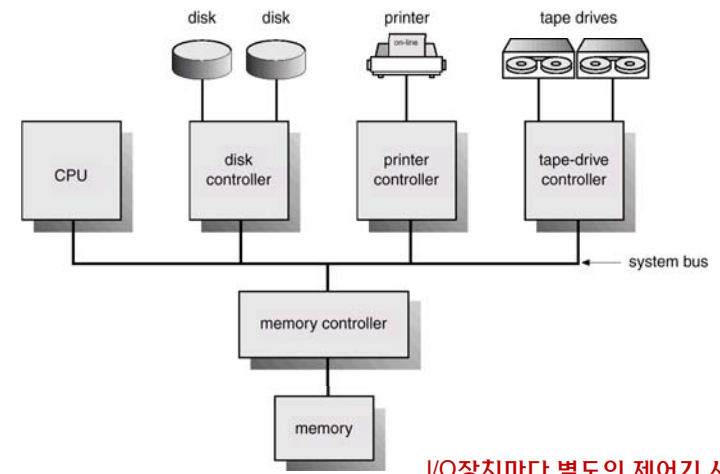


컴퓨터 시스템의 구성



컴퓨터시스템

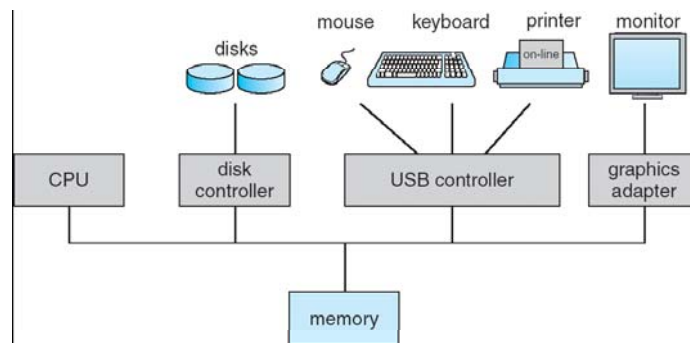
컴퓨터시스템 구성 예(1) - 예전



I/O장치마다 별도의 제어기 사용

컴퓨터시스템

컴퓨터시스템 구성 예(2) - 최근



여러 I/O장치를 연결할 수 있는 공용 제어기 사용

컴퓨터시스템

시스템 버스(1)

- 시스템 버스(system bus)
 - CPU와 시스템 내의 다른 요소들 사이에 정보를 교환하는 통로
 - 기본 구성
 - 주소 버스(address bus)
 - 데이터 버스(data bus)
 - 제어 버스(control bus)
- 주소 버스(address bus)
 - CPU가 외부로 발생하는 주소 정보를 전송하는 신호 선들의 집합
 - 주소 선의 수는 CPU와 접속될 수 있는 최대 기억장치 용량을 결정
 - [예] 주소 버스의 비트 수 = 16 비트라면, 최대 $2^{16} = 64K$ 개의 기억 장소들의 주소지정 가능

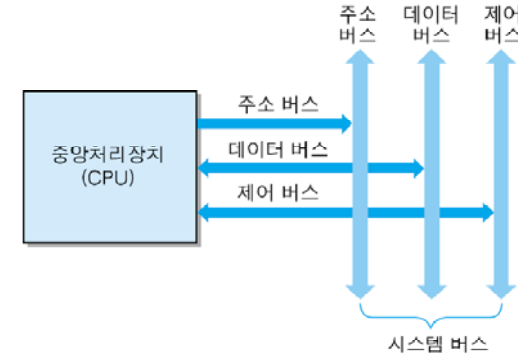
컴퓨터시스템

시스템 버스(2)

- 데이터 버스(data bus)
 - CPU가 기억장치 혹은 I/O 장치와의 사이에 데이터를 전송하기 위한 신호 선들의 집합
 - 데이터 선의 수는 CPU가 한 번에 전송할 수 있는 비트 수 결정
 - [예] 데이터 버스 폭 = 32 비트라면, CPU와 기억장치 간의 데이터 전송이 한 번에 32 비트씩 가능
- 제어 버스(control bus):
 - CPU가 시스템 내의 각종 요소들의 동작을 제어하기 위한 신호 선들의 집합
 - [예]
 - 기억장치 읽기/쓰기(Memory Read/Write) 신호
 - I/O 읽기/쓰기(I/O Read/Write) 신호

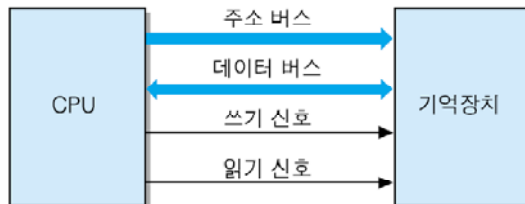
CPU와 시스템 버스 간의 접속

- 주소 버스 : 단방향성(uni-directional)
 - 주소는 CPU로부터 기억장치 혹은 I/O 장치들로 보내지는 정보
- 데이터 버스 : 양방향성(bi-directional)
 - 읽기와 쓰기 동작을 모두 지원



CPU와 기억장치 간의 접속

- 필요한 버스 및 제어신호
 - 주소 버스
 - 데이터 버스
 - 제어 신호: 기억장치 읽기(memory read) 신호, 기억장치 쓰기(memory write) 신호



기억장치 쓰기

- 기억장치 쓰기 동작 :
 - CPU가 데이터를 저장할 기억 장소의 주소와 저장할 데이터를 각각 주소 버스와 데이터 버스를 통하여 보내면서 동시에 쓰기 신호를 활성화
- 기억장치 쓰기 시간(memory write time) :
 - CPU가 주소와 데이터를 보낸 순간부터 저장이 완료될 때까지의 시간



(a) 기억장치 쓰기 동작의 시간 흐름도

기억장치 읽기

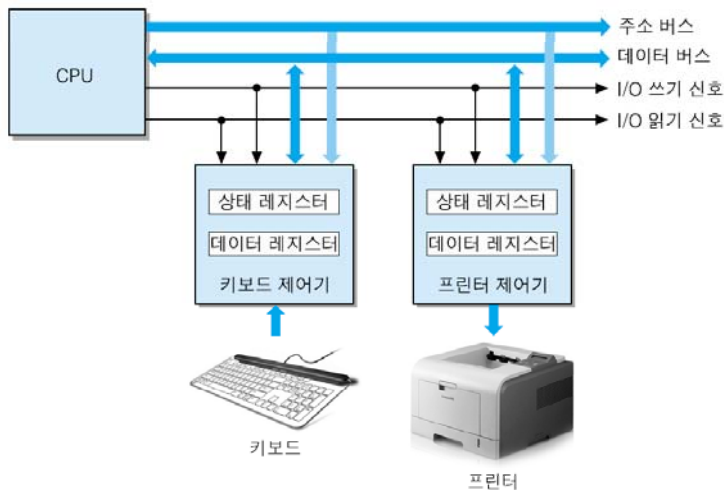
- 기억장치 읽기 동작
 - CPU가 기억장치 주소를 주소 버스를 통하여 보내면서 읽기 신호를 활성화
 - 일정 지연 시간이 경과한 후에 기억장치로부터 읽혀진 데이터가 데이터 버스 상에 실리며, CPU는 그 데이터를 버스 인터페이스 회로를 통하여 읽음
- 기억장치 읽기 시간(memory read time) :
 - 주소를 해독(decode)하는 데 걸리는 시간과 선택된 기억 소자들로부터 데이터를 읽는 데 걸리는 시간을 합한 시간



CPU와 I/O 장치의 접속

- 필요한 버스 및 제어 신호
 - 주소 버스
 - 데이터 버스
 - 제어 신호 : I/O 읽기(I/O read) 신호, I/O 쓰기(I/O write) 신호
- 접속 경로:
 - CPU - 시스템 버스 - I/O 장치 제어기 - I/O 장치

I/O 장치 접속 예 : CPU - 키보드&프린터



I/O 장치 제어기(I/O device controller)

- 기능:
 - CPU로부터 I/O 명령을 받아서, 해당 I/O 장치를 제어하고, 데이터를 이동함으로써 명령을 수행하는 전자회로 장치 (예) 키보드 제어기, 프린터 제어기 등
- 상태 레지스터
 - I/O 장치의 현재 상태를 나타내는 비트들을 저장한 레지스터
 - 준비 상태(RDY) 비트, 데이터 전송확인(ACK) 비트, 등
- 데이터 레지스터
 - CPU와 I/O 장치 간에 이동되는 데이터를 일시적으로 저장하는 레지스터

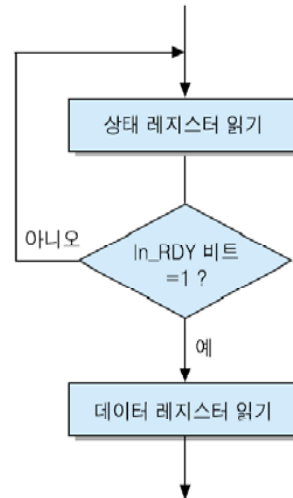
키보드 데이터 입력 과정

■ 키보드 제어기 :

- 키보드의 어떤 한 키(key)를 누르면, 그 키에 대응되는 ASCII 코드가 키보드 제어기의 데이터 레지스터에 저장되고, 동시에 상태 레지스터의 In_RDY 비트가 1로 세트

■ CPU :

1. 키보드 제어기로부터 상태 레지스터의 내용을 읽어서 In_RDY 비트가 세트되었는지 검사(In_RDY 비트는 데이터 레지스터에 외부로부터 데이터가 적재되었는지를 표시)
2. 만약 세트 되지 않았으면, 1번을 반복하며 대기. / 만약 세트 되었다면, 데이터 레지스터의 내용을 읽음



프린터 데이터 출력 과정

■ CPU :

1. 프린터 제어기의 상태 레지스터의 내용을 읽어서 Out_RDY 비트 검사 (Out_RDY 비트: 프린터가 출력할 준비가 되었는지를 표시)
2. 만약 세트 되지 않았으면, 1 번을 반복하며 대기 만약 세트 되었다면, 프린트할 데이터를 프린터 제어기의 데이터 레지스터에 저장

■ 프린터 제어기 :

- 데이터 레지스터의 내용을 프린터로 보내고,
- 프린터의 하드웨어를 제어하면서 인쇄 동작 수행

CPU와 보조저장장치의 접속

- 보조저장장치들(디스크, 플로피 디스켓, CD-ROM 등)도 각 장치를 위한 제어기를 통하여 키보드나 프린터와 유사한 방법으로 접속

■ 차이점 : 데이터 전송 단위

- 키보드, 프린터 : **바이트**(8 비트) 단위로 전송
- 보조저장장치 : **블록**(512/1024/4096 바이트) 단위로 전송
 - 제어기 내에 한 블록 이상을 임시 저장할 수 있는 데이터 버퍼 필요
 - **트랙 버퍼(track buffer)**: 하드 디스크상의 한 트랙 내용을 모두 저장할 수 있는 디스크 제어기 내의 데이터 버퍼

컴퓨터의 기본적인 기능들

■ 프로그램 실행 :

- CPU가 주기억장치로부터 프로그램 코드를 읽어서 실행

■ 데이터 저장 :

- 프로그램 실행 결과로서 얻어진 데이터를 주기억장치에 저장

■ 데이터 이동 :

- 디스크 혹은 CD-ROM에 저장되어 있는 프로그램과 데이터를 기억장치로 이동

■ 데이터 입력/출력 :

- 사용자가 키보드를 통하여 보내는 명령이나 데이터를 읽어 들인다. 또한 CPU가 처리한 결과값이나 기억장치의 내용을 프린터(혹은 모니터)로 출력

■ 제어 :

- 프로그램이 순서대로 실행되도록 또는 필요에 따라 실행 순서를 변경하도록 조정하며, 각종 제어 신호들을 발생

컴퓨터 구조의 발전 과정



- 주요 부품들의 발전 과정
 - 릴레이(relay) → 진공관 → 트랜지스터 → 반도체 집적회로(IC)
 - 발전 과정에서의 개선점들
 - 처리속도 향상
 - 저장용량 증가
 - 크기 감소
 - 가격 하락
 - 신뢰도 향상
- 초기 컴퓨터들의 근본적인 설계 개념과 동작 원리가 현대의 컴퓨터들과 거의 같음

최초의 컴퓨터



- 1642년, Blaise Pascal(프랑스)
- 덧셈과 뺄셈을 수행하는 기계적 카운터
- 다이얼의 위치에 의하여 십진수를 표시하는 6개의 원형판 세트들로 구성
- 각 원형판은 일시적으로 숫자를 기억하는 레지스터로 사용

Leibniz의 기계



- 1671년, Gottfried Leibniz(독일)
- 덧셈과 뺄셈 및 곱셈과 나눗셈도 할 수 있는 계산기
- Pascal의 계산기에 두 개의 원형판들을 추가하여 반복적 방법으로 곱셈과 나눗셈을 수행
- 이후 많은 기계들의 조상이 됨

Difference Engine



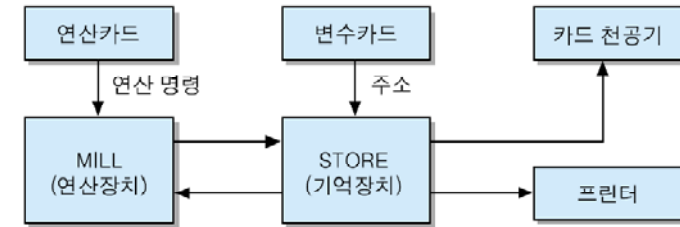
- 19세기 초, Charles Babbage(영국, 현대 컴퓨터의 할아버지)
- 표에 있는 수들을 자동적으로 계산하고, 그 결과를 금속천공기를 거쳐서 프린트
- 덧셈과 뺄셈만 수행 가능

Analytical Engine

- 19세기 초, Charles Babbage(영국)
- 주요 특징들
 - 어떤 수학 연산도 자동적으로 수행할 수 있는 일반목적용 계산기계
 - 프로그래밍 가능 : 프로그램 언어 사용
 - 프로그램의 실행 순서 변경 가능
 - 수의 부호 검사를 이용한 조건 분기
 - 제어카드를 이용한 실행 순서 변경
- 문제점
 - 주요 부품들이 기계적인 장치들이었기 때문에 속도가 느렸고 신뢰도가 낮음

Analytical Engine의 기본 구조

- 산술연산장치 : CPU
- 기억장치 : STORE
- 입력장치 : 카드판독기
- 출력장치 : 카드 천공기, 프린터



ENIAC

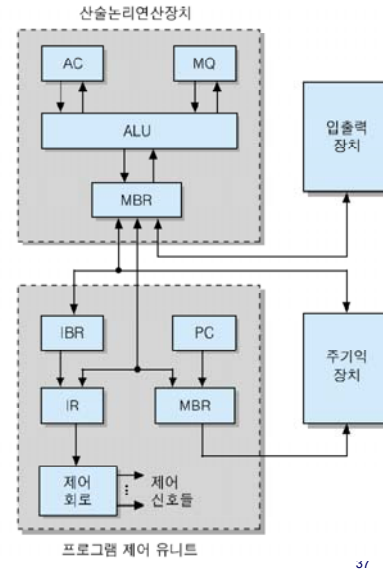
- Electronic Numerical Integrator And Computer
- 1940년대 초, von Neumann(폰 노이만)이 개발
- 펜실바니아 대학에서 개발한 진공관을 사용한 최초의 전자식 컴퓨터
- 문제점 : 프로그램의 저장과 변경 불가능
- 폰 노이만의 설계 개념(Stored-program 개념) 발표
 - 2진수 체계(binary number system)를 사용
 - 프로그램과 데이터를 내부에 저장
 - EDVAC(Electronic Discrete Variable Computer) 개발을 위하여 1945년에 발표

IAS 컴퓨터

- 1952년, 폰 노이만이 개발
- 'stored-program' 컴퓨터
- 폰 노이만의 설계 개념
 - 2진수 체계를 사용한다
 - 프로그램과 데이터를 내부에 저장한다
- 주요 구성요소
 - 프로그램 제어 유닛(Program Control Unit) : 명령어 인출/해독
 - 산술논리연산장치(ALU)
 - 주기억장치 : 명령어와 데이터를 모두 저장
 - 입출력장치

IAS 컴퓨터의 구조

- 폰 노이만 아키텍처(von Neumann Architecture):
프로그램 코드들을 기억장치에 저장된 순서대로 실행하며, 그 주소는 CPU의 내부 레지스터인 프로그램 카운터(program counter)에 의하여 지정



컴퓨터시스템의 분류와 발전 동향

- 개인용 컴퓨터
- 중형급 컴퓨터 시스템
- 메인프레임 컴퓨터
- 슈퍼 컴퓨터

- 임베디드 컴퓨터

개인용 컴퓨터(PC)

- 특징
 - 소형, 저가
 - 성능 : 십여년 전의 대형 메인프레임 컴퓨터의 성능을 능가
- 주요 발전 동향
 - 매 2~3년마다 성능이 개선된 새로운 마이크로프로세서가 등장하고, 그에 따라 새로운 PC 모델 출현
 - 주변 요소들(캐시, MMU, 산술보조프로세서 등)이 CPU 칩에 내장됨에 따라 속도 및 신뢰도가 향상
 - CPU 구조가 다수의 ALU들 혹은 명령어 실행 유니트들을 포함하는 슈퍼스칼라(superscalar) 구조, 듀얼-코어 및 쿼드-코어 구조로 발전

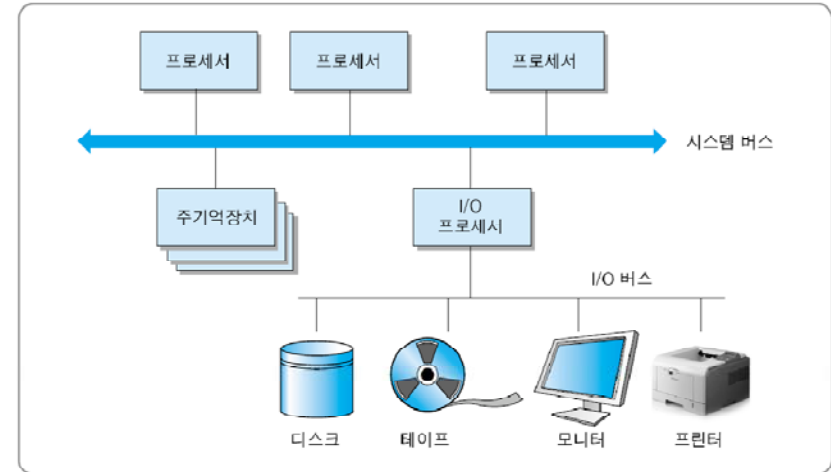
개인용 컴퓨터 (계속)

- 문자 이외의 다양한 정보들에 대한 입력과 출력, 저장 및 처리 능력을 보유하게 됨에 따라 멀티미디어 PC로 발전
- 보다 더 편리한 사용자 인터페이스를 제공해 주는 시스템 소프트웨어들 출현 (Windows 95/98/ME/2000/XP/Vista)
- 고속 I/O 장치들의 인터페이스를 위한 새로운 버스 규격 제안
- 주기억장치와 보조저장장치의 용량이 크게 증가, 종류 다양화

중형급 컴퓨터시스템

- 워크스테이션(workstation)
 - CPU : 64-비트 마이크로프로세서 사용
 - 고속 그래픽 처리 하드웨어 포함
 - 주요 응용 : 3차원 동영상처리, 시뮬레이션, CAD 등
 - OS : UNIX, LINUX
- 슈퍼미니컴퓨터(Super-minicomputer)
 - 시스템 구조 : **다중프로세서**(multiprocessor) 구조
 - CPU의 수 : 20 ~ 30 개
 - 성능 : VAX-11 미니컴퓨터 성능의 수십 배 이상
 - OS : UNIX
 - 서버(server)급 시스템의 다운사이징(downsizing) 주도
 - 네트워크에 접속된 다수의 중형급 컴퓨터 시스템들을 응용 (혹은 용도)별로 구분하여 사용하는 컴퓨팅 환경이 가능해지게 함

다중프로세서시스템의 구조



메인프레임(mainframe) 컴퓨터

- IBM 360 및 370 계열, 3081, 3090 등으로 계속 발전
- 대용량 저장장치 보유
- 다중 I/O 채널을 이용한 고속 I/O 처리 능력 보유
- 대규모 데이터베이스 저장 및 관리용으로 사용
- 최근 성능과 가격면에서 슈퍼미니급 컴퓨터들과 경쟁하고 있으며, 점차적으로 시장 점유율 하락 중

슈퍼컴퓨터(supercomputer)

- 현존하는 컴퓨터들 중에서 처리 속도와 기억장치 용량이 다른 컴퓨터들에 비하여 상대적으로 월등한 컴퓨터 시스템들
- 분류 기준 : 계속적으로 상승
 - 최초의 슈퍼컴퓨터인 CRAY-1의 속도는 100 MFLOPS
 - 최근의 슈퍼컴퓨터들의 속도는 TFLOPS급
- 주요 응용 분야들
 - VLSI 회로 설계, 항공우주공학, 천문학(일기 예보), 구조 공학, 유전 탐사, 핵공학, 인공지능, 입체 영상처리 등과 같은 대규모 과학 계산 및 시뮬레이션
- 구조적 특징에 따른 슈퍼컴퓨터 분류
 - 파이프라인 슈퍼컴퓨터(pipeline supercomputer)
 - 대규모 병렬컴퓨터(massively parallel computer: MPP)
 - 클러스터 컴퓨터(Cluster Computer)

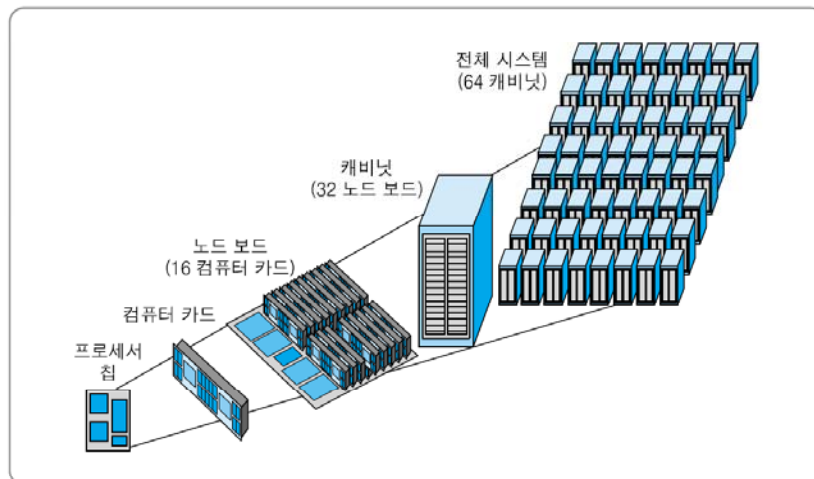
파이프라인 (pipeline)슈퍼컴퓨터

- 파이프라인 슈퍼컴퓨터(pipeline supercomputer)
 - 한 CPU 내에 다수의 연산 장치들이 포함
 - 각 연산 장치는 고도의 파이프라이닝 구조를 이용하여 고속 벡터 계산 가능
 - 대표적인 시스템들 : CRAY Y-MP, CRAY-2, Fujitsu VP2000, VPP500 등

대규모 병렬컴퓨터

- 대규모 병렬컴퓨터(massively parallel computer: MPP)
 - 한 시스템 내에 상호 연결된 수백 혹은 수천 개 이상의 프로세서들 포함
 - 프로세서들이 하나의 큰 작업을 나누어 동시에 처리하는 병렬처리(parallel processing) 기술 이용
 - 시스템 사례 : IBM BlueGene/L 슈퍼컴퓨터
 - 2008년 TOP500 리스트 최상위 랭크
 - 131,072 개의 64-비트 PowerPC 프로세서들 탑재
 - 64 캐비닛 x 32 노드 x 32 프로세서
 - 64 TByte 기억장치 보유

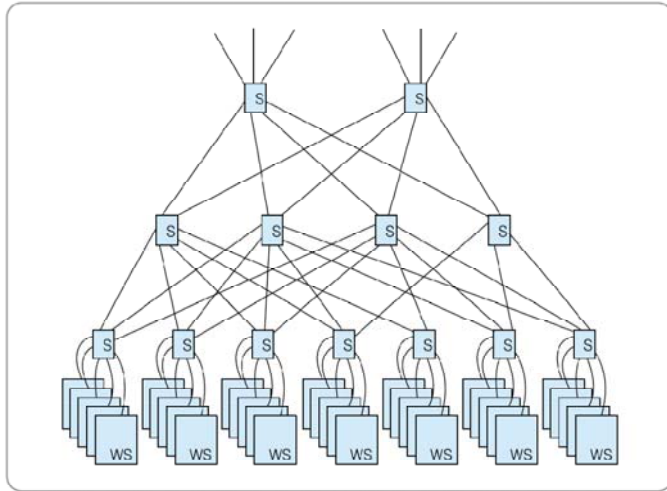
IBM BlueGene/L 슈퍼컴퓨터의 구성도



클러스터 컴퓨터

- 클러스터 컴퓨터(Cluster Computer)
 - 고속 LAN이나 네트워크 스위치에 의해 서로 연결된 PC들 혹은 워크스테이션들의 집합체
 - 노드(단위 컴퓨터)들에 포함된 모든 자원들을 단일 시스템 이미지(Single System Image: SSI)로 통합
 - 시스템 사례 : NOW(Network of Workstations)
 - U.C. 버클리대학 전산학과에서 개발
 - 105개의 워크스테이션들로 구성

NOW(Network of Workstations) 구성도



컴퓨터시스템

임베디드 컴퓨터

- Embedded Computer (내장 컴퓨터라고도 함)
- 기계 장치나 전자 장치들의 내부에 포함되어, 그 장치들의 동작을 제어(control)하는 컴퓨터들
 - [예] 가전제품, 컴퓨터 주변기기, 이동전화기, 비디오 게임기 등
- 8-비트 마이크로컨트롤러(micro-controller)를 이용한 초소형부터 32-비트 컴퓨터에 이르기까지 다양
- 최소의 비용으로, 필요한 만큼의 성능 제공
- 실시간 처리(real-time processing)
- 유비쿼터스 컴퓨팅 설비의 중심 요소로도 사용될 전망

컴퓨터시스템